



**You have downloaded a document from  
RE-BUS  
repository of the University of Silesia in Katowice**

**Title:** Aesthetic patterns from the perturbed orbits of discrete dynamical systems

**Author:** Krzysztof Gdawiec

**Citation style:** Gdawiec Krzysztof. (2013). Aesthetic patterns from the perturbed orbits of discrete dynamical systems. "Lecture Notes in Computer Science" (vol. 8104 (2013), pp. 358-366), doi 10.1007/978-3-642-40925-7\_33  
[postprint]



Uznanie autorstwa - Użycie niekomercyjne - Bez utworów zależnych Polska - Licencja ta zezwala na rozpowszechnianie, przedstawianie i wykonywanie utworu jedynie w celach niekomercyjnych oraz pod warunkiem zachowania go w oryginalnej postaci (nie tworzenia utworów zależnych).



UNIwersYTET ŚLĄSKI  
W KATOWICACH



Biblioteka  
Uniwersytetu Śląskiego



Ministerstwo Nauki  
i Szkolnictwa Wyższego

# Aesthetic Patterns from the Perturbed Orbits of Discrete Dynamical Systems

Krzysztof Gdawiec

Institute of Computer Science, University of Silesia, Poland  
kgdawiec@ux2.math.us.edu.pl

**Abstract.** The aim of this paper is to present some modifications of the orbits generation algorithm of discrete dynamical systems. The first modification is based on introduction of a perturbation mapping in the standard Picard iteration used in the orbit generation algorithm. The perturbation mapping is used to alter the orbit during the iteration process. The second modification combines the standard Picard iteration with the iteration which uses the perturbation mapping. The obtained patterns have unrepeatable structure and aesthetic value. They can be used for instance as textile patterns, ceramics patterns or can be used in jewellery design.

**Keywords:** dynamical system, orbit, perturbation mapping, aesthetic pattern

## 1 Introduction

One of the most elusive goals in computer aided design is artistic design and pattern generation. Pattern generation involves diverse aspects: analysis, creativity, development [15]. A designer has to deal with all of these aspects in order to obtain an interesting pattern which later could be used in jewellery design, carpet design, as a texture etc. Usually the most work during the design stage is carried out by a designer manually. Especially, in the case in which a graphic design should contain some unique unrepeatable artistic features. Therefore, it is highly useful to develop an automatic method for aesthetic pattern generation.

Aesthetics in the world of art and photography is connected with the principles of the nature and the perception of beauty [1]. Judging the beauty and other aesthetic qualities of patterns, paintings, photographs is a highly subjective task, so there is no standard method of measuring aesthetic values and it is a challenge to create such method in the emerging discipline of computational aesthetic [16]. However, in most of the works about pattern generation, in this paper also, the aesthetic is judged by the subjective feeling of the authors.

The literature is full of very diverse methods of pattern generation. For instance in [10] [14] the methods based on Iterated Function Systems and Genetic Algorithms for jewellery design were proposed. An interesting method based on root-finding polynomials, called polynomiography, was presented in [4]. In [13] a product-delay algorithm was proposed which is based on multiplication of sine

and square waves and in [5] authors presented a method which creates stone-like decorations using marbling.

Also very diverse methods based on discrete dynamical systems were proposed. In [8] authors made use of Gumowski-Mira transform for the generation of patterns used later as textile patterns. In their method they took the Gumowski-Mira transform and changed only the parameters of it. Other approach was presented in [2]. The authors have used different kinds of dynamical systems and using the switching process between the dynamical systems in conjunction with the Krasnoselskij iteration they generated aesthetic patterns. In this paper we propose another approach in obtaining aesthetic patterns from discrete dynamical systems. We modify the standard Picard iteration using the perturbation mapping and also combine both methods.

The paper is organized as follows. In Sect. 2 we introduce the basic information about discrete dynamical systems and present examples of dynamical systems which produce nicely looking orbits. Next, in Sect. 3 we introduce the notion of a perturbation mapping and how it can be used in the pattern generation. Some examples of aesthetic patterns obtained with the proposed methods are presented in Sect. 4. Finally, in Sect. 5 we give some concluding remarks.

## 2 Discrete Dynamical Systems

Let  $M$  be a subset in the  $q$ -dimensional Euclidean space  $\mathbb{R}^q$ . Following [9] discrete dynamical system is a continuous mapping  $\Phi : M \times \mathbb{N} \rightarrow M$  such that

$$\Phi(x, 0) = x, \quad (1)$$

$$\Phi(\Phi(x, t), s) = \Phi(x, t + s) \quad (2)$$

for all  $t, s \in \mathbb{N}$  and  $x \in M$ . The variable  $t$  is thought of as the time and generally discrete dynamical systems result from iterative processes or difference equations [9].

Assume that  $f : M \rightarrow M$  is a continuous mapping. Then  $f$  generates a discrete dynamical system of the form [9]:

$$\Phi(x, n) = f^n(x) = \underbrace{f \circ \dots \circ f}_{n \text{ times}}(x). \quad (3)$$

If  $n = 0$  then  $f^0(x) = x$ .

In the rest of the paper we will be interested in discrete dynamical systems given in a following form:

$$x_n = f^n(x) = f(f^{n-1}(x)) = f(x_{n-1}), \quad (4)$$

where  $n > 0$ . Formula (4) is also called the Picard iteration.

For our further considerations we also need the notion of an orbit. Orbit (or trajectory) of a point  $x_0$  is a sequence  $\{x_n\}_{n=0}^{\infty}$ , where  $x_n$  is given by (4).

Many examples of dynamical systems are known [7], but we are mainly interested in those which produce geometric patterns that can be recognized as aesthetic ones. Now, we present the examples of such dynamical systems in  $\mathbb{R}^2$ :

- Hopalong transformation [6]

$$\begin{aligned}x_n &= y_{n-1} - \operatorname{sgn}(x) \sqrt{|bx_n - c|}, \\y_n &= a - x_{n-1},\end{aligned}\tag{5}$$

where  $a, b, c \in \mathbb{R}$  and  $\operatorname{sgn} : \mathbb{R} \rightarrow \mathbb{R}$  is defined as follows:

$$\operatorname{sgn}(x) = \begin{cases} -1 & \text{if } x < 0, \\ 0 & \text{if } x = 0, \\ 1 & \text{if } x > 0, \end{cases}\tag{6}$$

- Zaslavsky transformation [7]

$$\begin{aligned}x_n &= (x_{n-1} + K \sin y_{n-1}) \cos \alpha + y_{n-1} \sin \alpha, \\y_n &= -(x_{n-1} + K \sin y_{n-1}) \sin \alpha + y_{n-1} \cos \alpha,\end{aligned}\tag{7}$$

where  $K \in \mathbb{R}$ ,  $\alpha = \frac{2\pi}{q}$ ,  $q \in \mathbb{N}$ ,  $q \geq 3$ ,

- Chip transformation created by Peters for the HOP program [11]

$$\begin{aligned}x_n &= y_{n-1} - \operatorname{sgn}(x_{n-1}) \cos(\ln |bx_{n-1} - c|)^2 \cdot \arctan(\ln |cx_{n-1} - b|)^2, \\y_n &= a - x_{n-1},\end{aligned}\tag{8}$$

where  $a, b, c \in \mathbb{R}$ ,

- Quadrup Two transformation also created by Peters for the HOP program [11]

$$\begin{aligned}x_n &= y_{n-1} - \operatorname{sgn}(x_{n-1}) \sin(\ln |bx_{n-1} - c|) \arctan(cx_{n-1} - b)^2, \\y_n &= a - x_{n-1},\end{aligned}\tag{9}$$

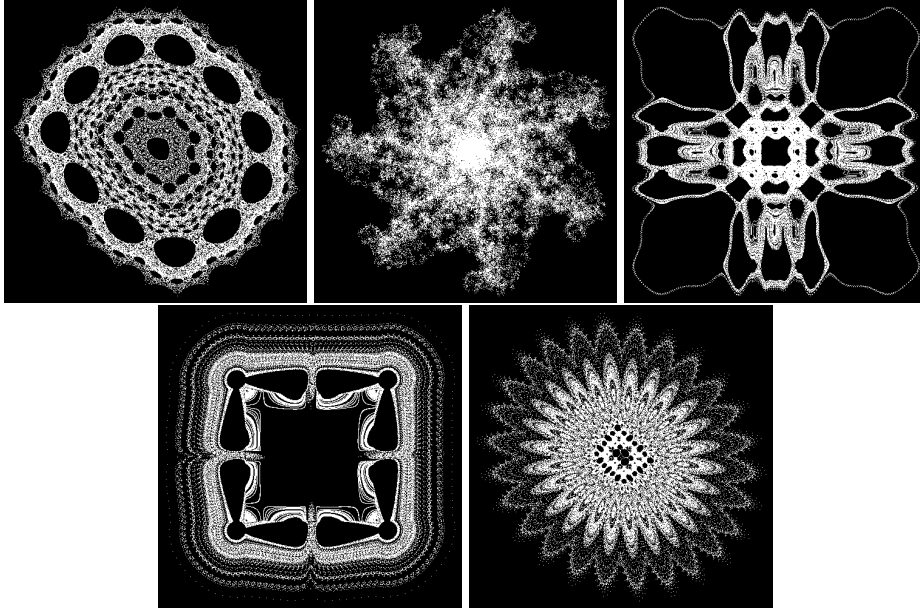
where  $a, b, c \in \mathbb{R}$ ,

- Three Ply transformation is another transformation created by Peters [11]

$$\begin{aligned}x_n &= y_{n-1} - \operatorname{sgn}(x_{n-1}) |\sin x_{n-1} \cos b + c - x_{n-1} \sin(a + b + c)|, \\y_n &= a - x_{n-1},\end{aligned}\tag{10}$$

where  $a, b, c \in \mathbb{R}$ .

The other examples of dynamical systems, producing interesting orbits, are: Gumowski-Mira transformation (CERN, 1980) [3], Martin [6], Cockatoo [6] etc. In Fig.1 the examples of orbits for the first 100000 points for transformations (5), (7), (8), (9), (10) are presented. The parameters for the transformations were following: Hopalong –  $a = 55$ ,  $b = -10$ ,  $c = -42.1$ ,  $x_0 = 0$ ,  $y_0 = 0$ , Zaslavsky –  $K = 4$ ,  $q = 7$ ,  $x_0 = 1$ ,  $y_0 = 1$ , Chip –  $a = 34$ ,  $b = 1$ ,  $c = 5$ ,  $x_0 = 0$ ,  $y_0 = 0$ , Quadrup Two –  $a = 34$ ,  $b = -1.2$ ,  $c = -20$ ,  $x_0 = 0$ ,  $y_0 = 0$ , Three Ply –  $a = 55$ ,  $b = -10$ ,  $c = -42.1$ ,  $x_0 = 0$ ,  $y_0 = 0$ .



**Fig. 1.** The examples of orbits ( $n = 100000$ ), the top row (from the left): Hopalong, Zaslavsky, Chip, the bottom row (from the left): Quadrup Two, Three Ply

### 3 Orbit Perturbation

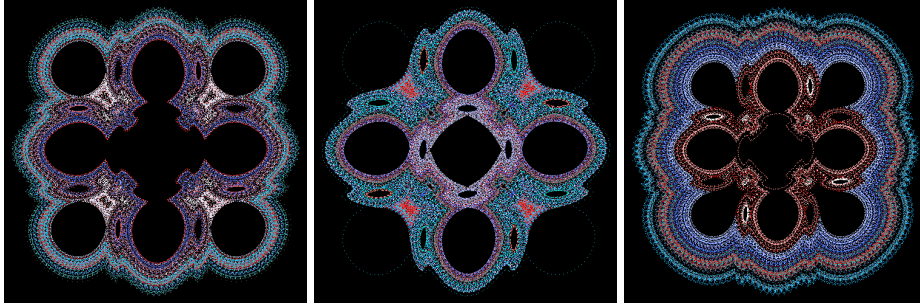
As we mention at the beginning of the paper there exist methods which use discrete dynamical systems presented in Sect. 2. They use changing of the transformation parameters or switching process in conjunction with the Krasnoselskij iteration. In our approach we are interested in the orbits of discrete dynamical systems.

Lets see an example of orbits for different starting points of a fixed dynamical system. Figure 2 presents orbits starting from (from left):  $(1.2, -2.05)$ ,  $(-3.55, -1.2)$  and  $(3.601, -1.05)$  for the Quadrup Two transformation with  $a = 34$ ,  $b = 1$ ,  $c = 5$  and  $n = 100000$ . To colour the orbits we used the iteration method [2] in which the colour of a point is given according to the number of iteration at which this point arises and a fixed colourmap. In this example we see that for different starting points we obtain various shapes of the orbits. This gives an idea that we can alter the orbits during the iteration process to change the shape of the final orbit.

We modify the Picard iteration given by (4) in a following way:

$$x_n = (f \circ p)^n(x) = (f \circ p)((f \circ p)^{n-1}(x)) = (f \circ p)(x_{n-1}), \quad (11)$$

where  $p : M \rightarrow M$  is a mapping. The mapping  $p$  is called a perturbation mapping and its aim is to alter (perturb) the orbit during the iteration process. We do not make any assumptions about the perturbation mapping because we can alter



**Fig. 2.** Orbits of Quadrup Two transformation for different starting points

the orbit in very different ways. When  $p(x) = x$  for all  $x \in M$  we do not have any perturbation of the orbit and therefore (11) reduces to (4).

A very simple example of perturbation mapping is:

$$p_v(x) = x + v, \quad (12)$$

where  $v \in M$ . This mapping simply translates the given point by  $v$ . In this way we change the orbit in every iteration in the same direction. We also can change the orbit by  $v$  only in some iterations, e.g. if the number of iteration is divisible by some fixed value.

Of course there is nothing in the way to use different values of  $v$  in consecutive iterations. In this case we can define some equation for  $v$  or we can define a grid with vectors and the appropriate vector is then computed using the bilinear interpolation. We also can give some randomness in the perturbation mapping taking in each iteration a random value of  $v$ .

When we have a perturbation mapping  $p$  we can define a new iteration process which combines the standard Picard iteration with the iteration given by (11). For instance this can be done in a following way:

$$x_n = \alpha x'_n + (1 - \alpha)x''_n, \quad (13)$$

where  $\alpha \in \mathbb{R}$  and

$$x'_n = f(x_{n-1}), \quad (14)$$

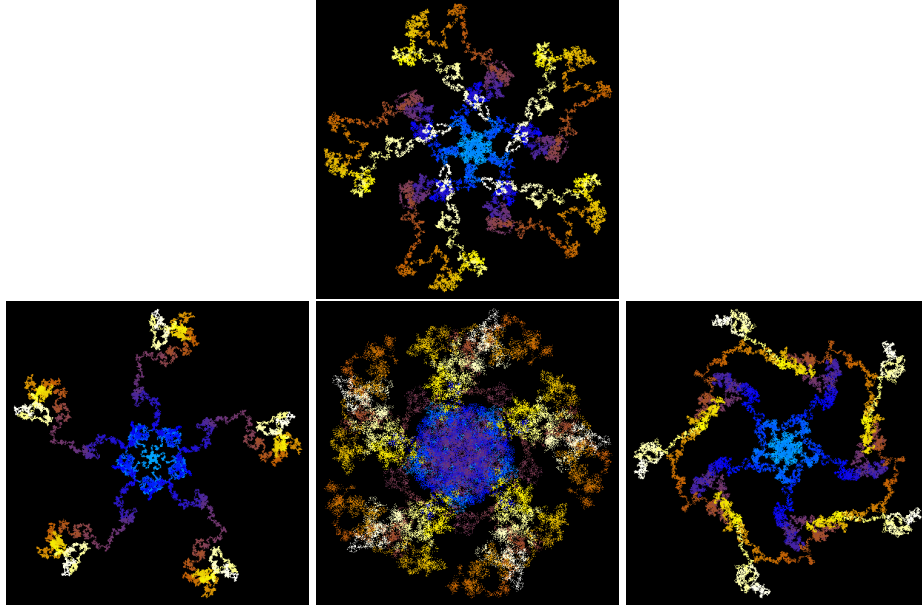
$$x''_n = f(p(x_{n-1})). \quad (15)$$

## 4 Examples

In this section we show some examples obtained using the methods presented in Sect. 3. In all the examples we use the same colouring method as in Fig. 2 but with other colourmaps.

We start with an example of Zaslavsky transformation with  $K = 3$ ,  $q = 5$ , starting point  $(1, 1)$  and number of iterations equal 100000. In the iteration

process we used  $p_v$  given by (12) with  $v = (1, 0.5)$  which is used after every: 500, 2500, 10000 iterations. The obtained results are presented in Fig. 3. The top part of the figure presents the original orbit without any perturbation, and the bottom part presents the patterns obtained with the perturbed orbits after every (from left): 500, 2500, 10000 iterations.

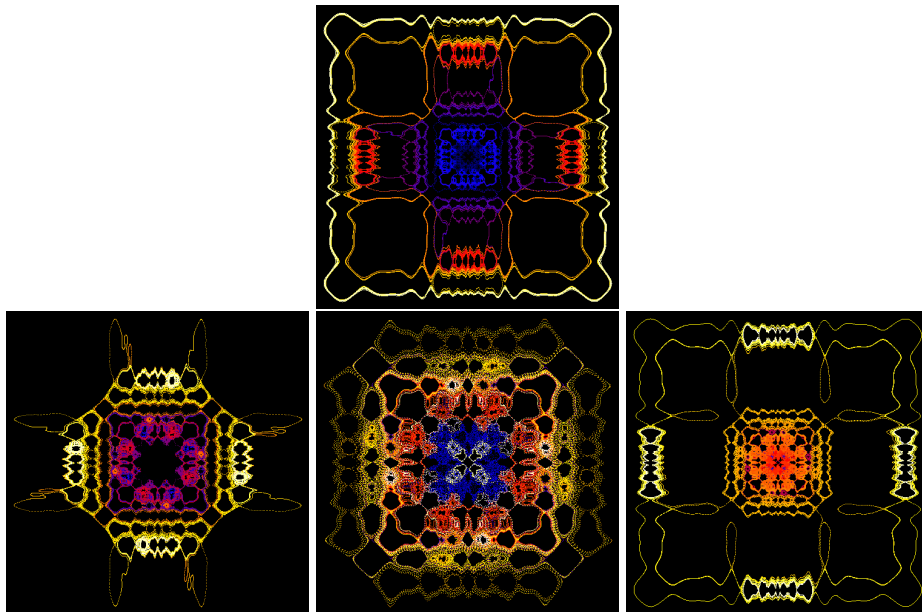


**Fig. 3.** Original orbit of Zaslavsky transformation (top) and patterns obtained with perturbation after a fixed number of iterations (bottom)

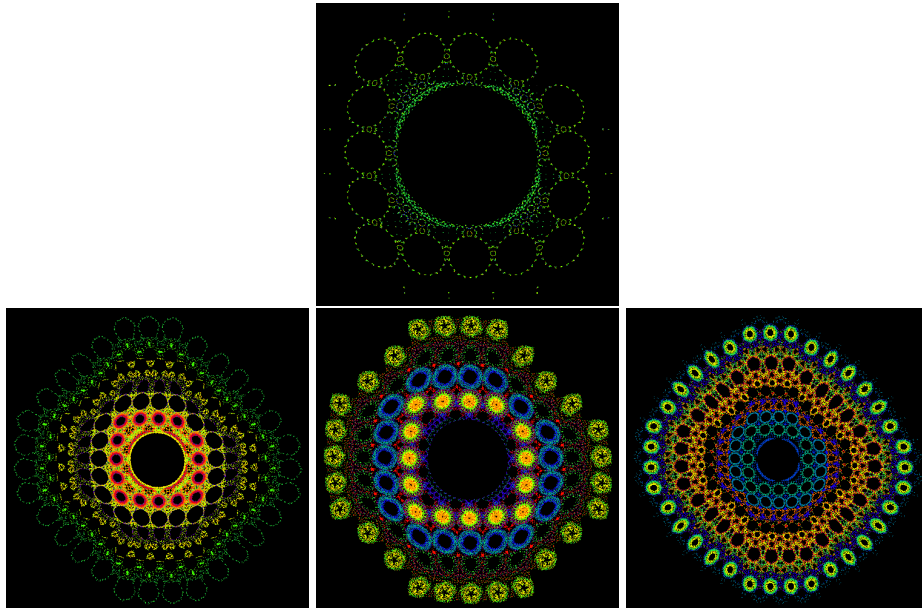
The next example presents the use of perturbation mapping  $p_v$  given by (12) with different values of  $v$ . This time we used the Chip transformation with  $a = 55$ ,  $b = -10$ ,  $c = -42$ , starting point  $(1, 1)$  and number of iterations equal 100000. Figure 4 presents the obtained patterns. The top part presents the original orbit, and the bottom part presents patterns for different values of  $v$  (from left):  $(0.1, 0.2)$ ,  $(9.1, -8.3)$ ,  $(-0.58, 2.16)$ .

Using the random vectors in the perturbation mapping we are also able to obtain very interesting and diverse patterns which shows next example. For this example we used the Hopalong transformation with  $a = -55$ ,  $b = -1$ ,  $c = -42$ , starting point  $(0, 0)$  and number of iterations equal 100000. The obtained patterns are presented in Fig. 5, where the original orbit is in the top part, and in the bottom part we have patterns obtained with the random vectors. The co-ordinates of the random vector were randomly chosen from  $[-0.05, 0.05]$ .

In the last example we present some patterns obtained using the combination of the Picard iteration and the perturbed iteration. We used the Three Ply



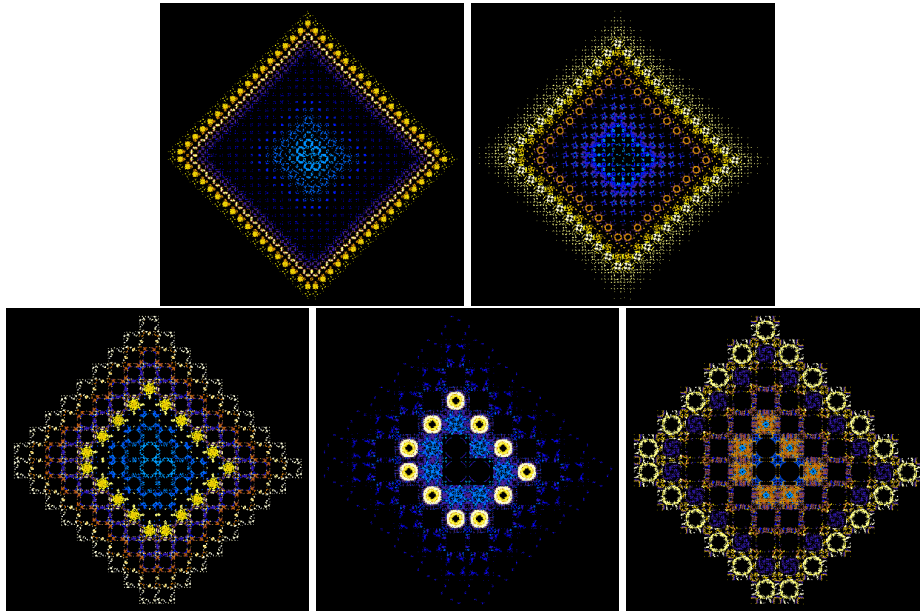
**Fig. 4.** Original orbit of Chip transformation (top) and patterns obtained with perturbation using different vectors (bottom)



**Fig. 5.** Original orbit of Hopalong transformation (top) and patterns obtained with perturbation using random vectors (bottom)



transformation with  $a = -54.4$ ,  $b = -1$ ,  $c = -42$ , starting point  $(0, 0)$  and number of iterations equal 100000. The obtained patterns are presented in Fig. 6. The original orbit is in top-left part of the figure, and in the top-right we have a pattern obtained with perturbation with a fixed vector  $v = (-0.5, 0.5)$ . In the bottom part we have the patterns obtained with the combination method with different values of  $\alpha$  (from left): 0.3, 0.6, 0.9.



**Fig. 6.** Original orbit of Three Ply transformation (top-left), pattern obtained with perturbation mapping with a fixed vector (top-right) and patterns obtained using the combination (13) with different values of  $\alpha$  (bottom)

## 5 Conclusions

In the paper we presented new methods for patterns generation with the use of discrete dynamical systems. The first method uses the perturbation mapping, which alters the orbit of a dynamical system during the iteration process. We also presented some perturbation mappings. The second method uses a combination of the two methods: the standard Picard iteration and the method with the perturbation mapping.

The presented examples show that using the proposed methods we are able to obtain very interesting and diverse patterns. The patterns differ from those obtained with the standard Picard iteration. Comparing to the method from [8] the proposed methods give more possibilities of obtaining unrepeatable patterns.

The obtained patterns have an aesthetic value so they can be used as usable patterns, e.g. textile patterns, ceramics patterns, or can be used in jewellery, decoration design or in generating textures using for instance the glyph bombing technique [12].

## References

1. Datta, R., Joshi, D., Li, J., Wang, J.Z.: Studying Aesthetics in Photographic Images Using a Computational Approach. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006, Part III. LNCS, vol. 3953, pp. 288–301. Springer, Heidelberg (2006)
2. Gdawiec, K., Kotarski, W., Lisowska, A.: Automatic Generation of Aesthetic Patterns with the Use of Dynamical Systems. In: Bebis, G., et al. (eds.) ISVC 2011, Part II. LNCS, vol. 6939, pp. 691–700. Springer, Heidelberg (2011)
3. Gumowski, I., Mira, C.: Recurrences and Discrete Dynamic Systems. Springer, New York (1980)
4. Kalantari, B.: Polynomial Root-Finding and Polynomiography. World Scientific, Singapore (2009)
5. Lu, S., Jaffer, A., Jin, X., Zhao, H., Mao, X.: Mathematical Marbling. IEEE Computer Graphics and Applications 32(6), 26–35 (2012)
6. Martin, B.: Graphic Potential of Recursive Functions. In: Landsdow, J., Earnshaw, R.A. (eds.) Computers in Art, Design and Animation, pp. 109–129. Springer, Heidelberg (1989)
7. Morozov, A.D., Dragunov, T.N., Boykova, S.A., Malysheva, O.V.: Invariant Sets for Windows. World Scientific, Singapore (1999)
8. Naud, M., Richard, P., Chapeau-Blondeau, F., Ferrier, J.L.: Automatic Generation of Aesthetic Images for Computer-assisted Virtual Fashion Design. In: Proceedings 10th Generative Art Conference, Milan, Italy (2007)
9. Osipenko, G.: Dynamical Systems, Graphs, and Algorithms. Springer, New York (2007)
10. Pang, W., Hui, K.C.: Interactive Evolutionary 3D Fractal Modeling. Visual Computer 26(12), 1467–1483 (2010)
11. Peters, M.: HOP – Fractals in Motion, <http://www.mpeters.de/mpeweb/hop/>
12. Rost, R.J., Licea-Kane, B.: OpenGL Shading Language, 3rd edn. Addison-Wesley, Boston (2010)
13. Sen, A.K.: A Product-Delay Algorithm for Graphic Design. Computers & Graphics 22(6), 759–764 (1998)
14. Wannarumon, S., Bohez, E.L.J.: A New Aesthetic Evolutionary Approach for Jewelry Design. Computer-Aided Design & Applications 3(1-4), 385–394 (2006)
15. Wannarumon, S., Unnanon, K., Bohez, E.L.J.: Intelligent Computer System for Jewelry Design Support. Computer-Aided Design & Applications 1(1-4), 551–558 (2004)
16. Wu, Y., Bauckhage, C., Thureau, C.: The Good, the Bad, and the Ugly: Predicting Aesthetic Image Labels. In: Proceedings 20th International Conference on Pattern Recognition, Istanbul, Turkey, pp. 1586–1589 (2010)